

УДК 338.28

# Анализ методик оценки трудоемкости разработки программного обеспечения

В статье представлены итоги научного исследования, результатами которого являются анализ методик оценки трудоемкости разработок программного обеспечения, выявление их преимуществ и недостатков, определение наиболее приемлемой для применения методики с учетом адаптации для белорусской практики формирования государственного заказа на выполнение разработок в сфере информационных технологий.

**А.В. БУДНИК,**кандидат технических наук, доцент,  
декан факультета инжиниринга и технологий связи**О.В. ДОМАКУР,**кандидат экономических наук,  
доцент, ученый секретарь**Е.С. РОМАНОВА,**кандидат экономических наук, доцент,  
доцент кафедры цифровой экономики**Т.Л. ТРУХАНОВИЧ,**старший преподаватель кафедры  
программного обеспечения сетей телекоммуникаций

УО «Белорусская государственная академия связи»

**Ключевые слова:***трудоемкость, программное обеспечение, метрики, методики, оценка затрат труда.*

**Введение.** В период развития нового вида деятельности цены на продукт производства определяются, прежде всего, спросом на него, который увеличивается по мере ознакомления с преимуществами потребления нового продукта. В период зрелости, когда производственные процессы стандартизируются, появляются основания для сравнения и объективной оценки реальных затрат на производство продукта.

Сегодня рыночная цена программного продукта представляет собой в большей степени оценку возможностей заказчика оплатить разработку. Однако наступил момент, когда возникала необходимость более объективной оценки создания программного обеспечения.

В условиях асимметрии информации и неубедительности обоснований стоимости работ по созданию программного обеспечения для принятия решения о заказе разработки органы государственного управления должны иметь инструментарий для объективного обоснования стоимости программного обеспечения. Данный инструментарий должен позволять разработчику выполнить оценку всех своих затрат на основе трудоемкости работ по его созданию, обладать прозрачностью для заказчика, который без использования специфических знаний и конфиденциальной информации, сможет обосновать приемлемую цену сделки.

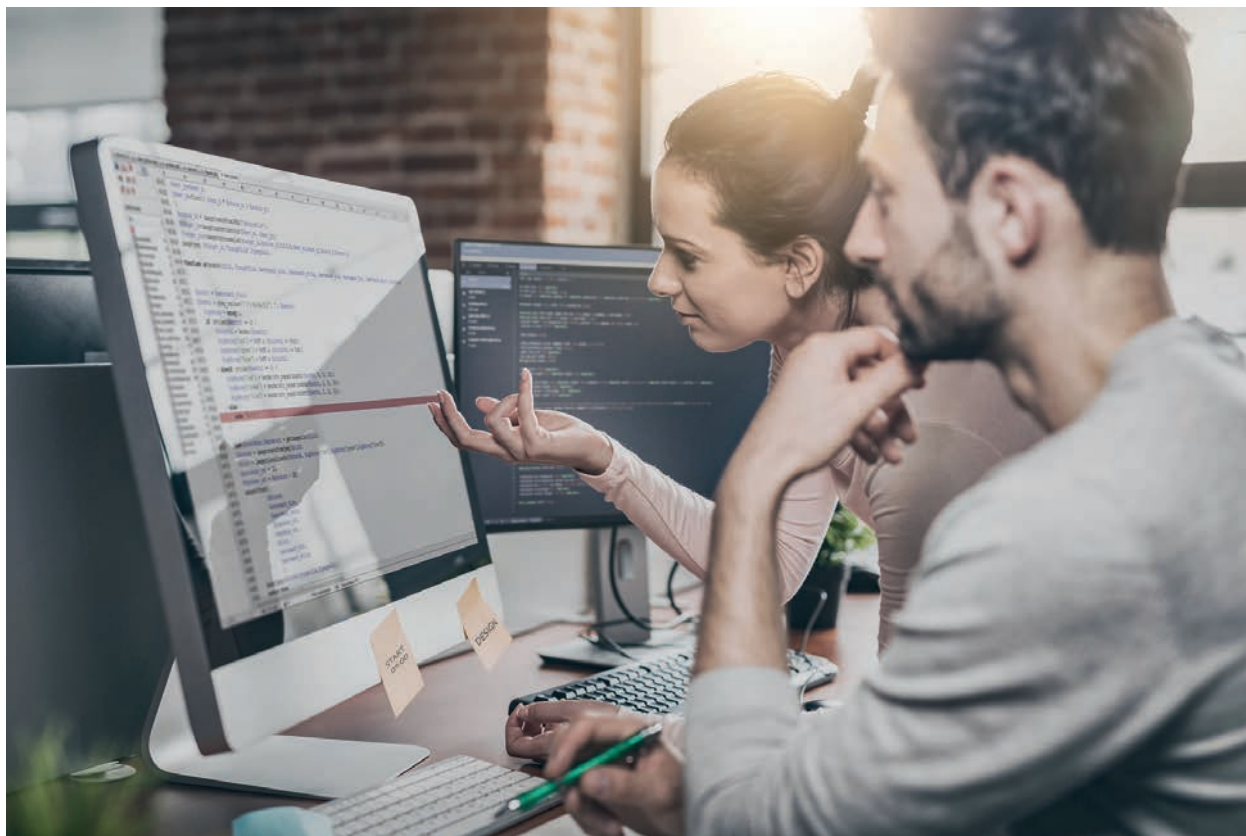
В статье представлены результаты поиска инструментария объективной оценки трудоемкости разработки программного обеспечения, учитывающего современные способы программирования, методы разработки и нормы труда, а также возможность расчета стоимости разработки ПО.

**Метрики оценки размеров программного обеспечения.** Метод оценки размера ПО IFPUG основан на предположении о связи между языком программирования и количеством строчек кода, приходящимся на одну функцию. Оценка количества функций, приходящихся на реализацию одного функционального требования, учитывает его сложность, а количество полученных функциональных точек пересчитывается в условные строчки кода, или в трудозатраты. Функциональные точки остаются постоянными и не зависят от квалификации программистов или языка программирования.

Метод оценки размера ПО IFPUG предполагает последовательное прохождение этапов.

Этап 1. Определение типа объекта для оценки функциональных точек: проект разработки, проект развития, готовый продукт.

Этап 2. Определение области оценки и границ продукта. В зависимости от типа область оценки может включать: все разрабатываемые функции (для проекта разработки); все добавляемые, изменяемые и удаляемые функции (для проектов



поддержки); только функции, реально используемые, или все функции (при оценке продукта и/или продуктов).

Этап 3. Подсчет функциональных точек, связанных с данными. На данном этапе сначала определяется сложность данных в системе координат по показателям неповторяемое уникальное поле данных (например, Имя Клиента – 1 DET; Адрес Клиента (индекс, страна, область, район, город, улица, дом, корпус, квартира) – 9 DET's) и логическая группа данных (например, адрес, паспорт, телефонный номер).

Этап 4. Подсчет функциональных точек, связанных с транзакциями (процесс перевода ПО из одного состояния в другое) и оценка сложности транзакции

Этап 5. Определение суммарного количества не выровненных функциональных точек путем суммирования по всем информационным объектам и транзакциям.

Этап 6. Определение значения фактора выравнивания по шкале от 0 до 5 баллов по параметрам системных характеристик продукта: обмен данными, распределенная обработка данных, производительность, ограничения по аппаратным ресурсам, транзакционная нагрузка, интенсивность взаимодействия с пользователем, эргономика, интенсивность изменения данных пользователями,

сложность обработки, повторное использование, удобство инсталляции, удобство администрирования, портируемость, гибкость. Расчет суммарного эффекта системных характеристик

Расчет значения фактора выравнивания производится по формуле

$$VAF = (TDI \times 0,01) + 0,65, \quad (1)$$

где VAF (value of alignment factor) – фактор выравнивания;

TDI (total degree of influence) – суммарный эффект характеристик разрабатываемого ПО;

0,01 – коэффициент выравнивания, рассчитанный на основе предварительных оценок;

0,65 – коэффициент выравнивания, рассчитанный на основе предварительных оценок.

Этап 7. Расчет количества выровненных функциональных точек определяется умножением количества не выровненных функциональных точек на фактор выравнивания. Необходимо дополнять не выровненные функциональные точки функциональностью для установки продукта, для измененных функций, рассчитанные после модификации.

Суммарное влияние процедуры выравнивания лежит в пределах  $\pm 35\%$  относительно объема рассчитанного в не выровненных функциональных точках [1].

Таким образом, для оценки трудоемкости по методике IFPUG необходимо достаточно детально

смоделировать структуру ПО, выполняемые в нем функции, количество интерфейсов, структуру данных и атрибуты данных.

Метод оценки размера ПО UCP базируется на выявлении вариантов использования (Use Cases Points) пользователем функционала ПО, представляет собой описание взаимодействия пользователя и ПО и разработку функциональных решений желаний пользователей.

Оценка по методу UCP складывается из следующих этапов:

1) оценка акторов;  
2) нескорректированная оценка вариантов использования;

3) оценка технических факторов;

4) оценка внешних факторов;

5) оценка трудоемкости проекта по формуле:

$$UCP = (UUCW + UAW) \times TCF \times ECF, \quad (2)$$

где UUCW – нескорректированная оценка вариантов использования;

UAW – оценка акторов;

TCF – оценка технических факторов;

ECF – оценка внешних факторов [2].

При использовании метода UCP как метрики размера проекта вычисления на этом заканчиваются, для оценки трудоемкости требуется перевести полученный результат из количества вариантов использования в трудозатраты.

**Методики оценки трудоемкости разработки ПО.** Методика СОСОМО основана на базе данных о проектах и представляет собой оценку по этапам разработки требований к ПО: базовая (Basic) применяется на этапе выработки спецификаций требований; расширенная (Intermediate) – после определения требований к ПО; углубленная (Advanced) – после окончания проектирования ПО.

В общем виде, уравнение моделей имеет вид:

$$E = a \times S^b \times EAF, \quad (3)$$

где  $E$  – затраты труда на проект (в человеко-месяцах);

$S$  – размер кода (в KSLOC);

$EAF$  – фактор уточнения затрат (effort adjustment factor);

$a$  – коэффициент, рассчитанный на основе предварительных оценок;

$b$  – коэффициент, рассчитанный на основе предварительных оценок.

Параметры  $a$  и  $b$  зависят от сложности разрабатываемого приложения: простое прикладное ПО; ПО средней сложности (например, система управления банковским терминалом); сложное ПО (например, система управления полетами) [3].

Для определения размера кода продукта можно использовать три способа оценки: объектные точки, функциональные точки, количество логических строк кода. Число объектных и функциональных точек пересчитываются в количество строк кода.

В методике СОСОМО учитываются факторы масштаба проекта: наличие опыта аналогичных разработок; гибкость процесса; архитектура и разрешенные риски; сработанность команды; зрелость процессов. Также учитываются такие факторы как квалификация персонала; опыт персонала; сложность и надежность продукта; возможность повторного использования; сложность платформы разработки; оборудование, инструменты простейшие/интегрированные, средства поддержки жизненного цикла; требуемое выполнение графика работ.

Таким образом, метод СОСОМО позволяет получить оценку трудоемкости разработки ПО с учетом масштаба ПО, факторов трудоемкости, а также условий разработки.

Методика Постановления Министерства труда и социальной защиты Республики Беларусь 27.06.2007 № 91, действующая в настоящее время в Республике Беларусь для определения общей трудоемкости разработки программного обеспечения для организаций всех форм собственности, рекомендует использовать укрупненные нормы затрат труда, в основе которых заложены результаты анализа фактических затрат труда на разработку ПО, экспертные оценки, данные оперативного учета и отчетности, результаты анализа действовавших ранее и в настоящее время норм труда по разработке ПО в Российской Федерации и других странах.

Основными факторами, влияющими на трудоемкость разработки ПО, определены: объем ПО в строках исходного кода; сложность разрабатываемого ПО; степень новизны разрабатываемого ПО; степень использования в разработке стандартных модулей; условия и средства разработки ПО.

Общий объем ( $V_o$ ) программного продукта определяется исходя из количества и объема функций, реализуемых программой. В зависимости от организационных и технологических условий, в которых разрабатывается ПО, объем корректируется на основе экспертных оценок.

По уточненному объему ПО и нормативам затрат труда в расчете на единицу объема определяются нормативная и общая трудоемкость разработки ПО.

Нормативная трудоемкость определяется по стадиям разработки проекта (техническое задание, эскизный проект, технический проект, рабочий проект, ввод в действие) и может корректироваться при необходимости с учетом коэффициентов повышения сложности ПО, учитывающих новизну ПО,



учитывающих степень использования стандартных модулей, средства разработки ПО.

Скорректированная нормативная трудоемкость по стадиям разработки в дальнейшем служит основой для определения общей трудоемкости [4].

Методика SETIN основана на подсчете числа функциональных требований пользователей к разрабатываемому ПО с помощью пяти единиц измерения размера ПО, от заглавных букв которых и происходит ее название.

Методика содержит алгоритмы расчета

- 1) размера ПО;
- 2) трудоемкости разработки ПО;
- 3) стоимости разработки и сопровождения ПО.

Порядок оценки трудоемкости разработки ППО представлен следующими этапами:

- 1) оценка функционального размера разрабатываемой ИС;
- 2) оценка базовой трудоемкости разработки ПО;
- 3) определение значений поправочных коэффициентов трудоемкости разработки и сопровождения ПО, учитывающие факторы разработки и эксплуатации ПО;
- 4) расчет трудоемкости разработки ПО с учетом поправочных коэффициентов;
- 5) оценка срока разработки ПО;
- 6) корректировка трудоемкости разработки ПО при уменьшении срока разработки;
- 7) оценка стоимости разработки ПО.

Этап 1. Оценка функционального размера. Оценка функционального размера информационной системы производится на основании модели информационной системы и функциональных требований пользователей. Функциональный размер ИС представляет собой пять функциональных единиц измерения: (С – Case, E – Entity, T – Tool, I – Interaction, N – Node):

- количество вариантов использования – С;
- количество типов объектов – E;
- количество свойств типов объектов – T;
- количество взаимодействий между типами объектов – I;
- количество типов узлов – N.

Функциональный размер обозначается как SIZE = {C, E, T, I, N}.

Для оценки функционального размера рекомендуется использовать модель ПО, разработанной на языке моделирования UML. В случае отсутствия такой модели оценщику предлагается заполнить анкету с перечнем вопросов о ПО для определения его функциональности на ранних этапах, оценить границы системы.

Этап 2. Оценка базовой трудоемкости на каждом этапе процесса разработки ПО: бизнес

моделирование; управление требованиями; проектирование; разработка; тестирование; внедрение.

Базовая трудоемкость каждого процесса разработки рассчитывается как сумма произведений единиц измерения функционального размера и значений показателей трудоемкости соответственно.

Трудоемкость реализации одного варианта использования, одного типа объектов, одного свойства типа объекта, одного взаимодействия между типами объектов, одного типа узла в процессе разработки рассчитывается по нормативам.

Этап 3. Выявление технических требований к объекту информатизации, требований качества пользователя, выбор поправочных коэффициентов.

Этап 4. Расчет трудоемкости с учетом поправочных коэффициентов по предложенной таблице.

Этап 5. Оценка срока разработки информационной системы по предложенной таблице.

Этап 6. Корректировка трудоемкости разработки ПО при уменьшении срока разработки. В методике предусмотрен механизм расчета трудоемкости при уменьшении срока разработки, который основан на математической модели эластичности трудоемкости [5].

Основным отличием метода SETIN от описанных выше методов IFPUG и СОСОМО II является возможность его применения в условиях отсутствия подробного технического задания или спецификации будущего ПО, при наличии только концепции разрабатываемого ПО.

Таким образом, методика SETIN позволяет оценить функциональный размер ПО, учитывает современные методы объектно-ориентированного программирования и этапы разработки ПО, применима на ранних этапах разработки, при различных моделях жизненного цикла разработки.

**Анализ метрик оценки размера ПО и методик оценки трудоемкости разработки ПО.** Методика оценки функционального размера ПО IFPUG предполагает наличие детального описания структуры ПО, выполняемых функций, количества интерфейсов, структуры данных и атрибутов данных. Достоинствами данной методики является возможность достаточно точно и объективно оценить размер ПО разных видов без привязки к таким факторам, как квалификация программистов или язык программирования, возможна оценка на ранних этапах разработки ПО, применима к любой модели жизненного цикла разработки ПО. Недостатками данной методики являются высокая трудоемкость оценки, необходимость экспертных знаний в сфере проектирования и функционирования ПО, точное определение функционала ПО, видов данных, способов их ввода, передачи, обработки, вывода, применима

при наличии у компании разработчика собственной статистики трудозатрат на реализацию функциональных точек.

Метод UCP является метрикой размера ПО. Достоинством метода UCP является возможность его использования при использовании современной методологии RUP. Метод UCP отличается простотой расчетов, возможностью оценки на ранних этапах разработки ПО без привлечения экспертов, т. к. не требуется описания точного функционала ПО или подробной спецификации требований. Методика UCP универсальна, отсутствует требование учета специфики языков программирования и моделей разработки ПО. Недостатками метода UCP являются учет внешних факторов, мало относящиеся к объективной оценке трудоемкости разработки ПО, например, мотивация проектной команды и удобство установки разработанного ПО. Отсутствие оценки количества затрат труда на разработку отдельных элементов, выделенных в качестве единицы размера ПО, в данном методе, приводит к выводу о невозможности использования его в данной работе.

Методика СОСОМО II позволяет получить вероятностную оценку трудоемкости разработки ПО, с учетом групп факторов трудоемкости и масштаба ПО. Достоинствами методики СОСОМО II являются простота применения, наличие коэффициентов пересчета размера ПО в трудоемкость работ. Данная методика учитывает особенности языков программирования, подходит для любых моделей жизненного цикла разработки ПО.

Недостатками методики СОСОМО II являются учет внешних факторов, мало относящиеся к объективной оценке трудоемкости разработки ПО,

например, опыт и квалификация персонала, работоспособность команды, инструменты разработки (простейшие/интегрированные), средства поддержки жизненного цикла, гибкость процессов, методика СОСОМО II применима для ограниченного числа языков программирования, ее использование требует специфических экспертных знаний языков программирования для подсчета количества строк кода, глубокой предварительной проработки архитектуры ПО. Коэффициенты, применяемые в СОСОМО II, рассчитывались на основе статистики проектов, разрабатываемых в 1997 году, поэтому возникают сомнения в актуальности их применения на современном этапе развития языков программирования и методологий разработки ПО.

Методика определения общей трудоемкости разработки программного обеспечения для организаций всех форм собственности, утвержденная Постановлением Министерства труда и социальной защиты Республики Беларусь 27.06.2007 № 91 проста в применении, содержит обоснованные нормы труда для расчета трудоемкости и затрат на разработку ПО.

Недостатки методики: требуется детальная проработка требований и архитектуры проектируемого ПО, необходимо привлечение экспертов, невозможно применение на ранних стадиях разработки ПО, не в полной мере соответствует современным моделям разработки ПО, обобщен опыт разработки проектов до 2007 года, не учитывает особенности этапов разработки ПО (стадии разработки больше соответствуют стадиям разработки нового изделия для промышленного производства), применима для ограниченного числа языков программирования, учитывает внешние факторы, мало относящиеся



к объективной оценке трудоемкости разработки ПО, например, применяемые средства разработки. Нормативная трудоемкость рассчитана в человеко-днях, однако более гибким измерителем считается количество человеко-часов. При этом примерные нормы могут корректироваться разработчиками, в то время как заказчик ПО может не знать всех нюансов разработки и часто не способен применить данную методику.

Методика SETIN основана на оценке функционального размера ПО и позволяет оценить трудоемкость разработки ПО на этапе концепции. Важно наличие спецификации будущего ПО, ее качество повышает точность оценки. Достоинствами методики являются объективность оценки трудоемкости разработки функциональных единиц, учет современных методов объектно-ориентированного программирования и этапов разработки программного обеспечения. Недостатки – необходимость ознакомления специалистов заказчика с алгоритмом оценки функционального размера ПО, наличие качественно выполненной спецификации требований к ПО до заключения договора на выполнение работ по разработке ПО.

**Заключение.** В целом можно заключить, что методики оценки размеров ПО (IFPUG, UCP) требуют дополнений в части оценки количества затрат труда на разработку отдельных элементов, выделенных в качестве единицы размера ПО. Поэтому они не могут быть полноценно использованы в данной работе либо потребуют сбора большого массива статистических данных о затратах труда в организациях,

занимающихся разработкой, их усреднения и апробации для применения в полноценной методике.

Методика СОСОМО II и Методика определения общей трудоемкости разработки программного обеспечения для организаций всех форм собственности, утвержденная Постановлением Министерства труда и социальной защиты Республики Беларусь 27.06.2007 № 91, морально устарели, имеют много ограничений в применении заказчиком ПО, учитывают внешние факторы, которые заказчику почти невозможно оценить, особенно в случае отсутствия претендентов на выполнение заказа по разработке ПО. Кроме того, при выполнении какого-либо заказа в других видах экономической деятельности исполнитель самостоятельно определяет состав работников (их квалификацию и мотивацию), способных выполнить работы, а также перечень оборудования и инструментов, необходимых для ее выполнения, чтобы удовлетворить требования заказчика по качеству продукта или работ, срокам исполнения и суммам оплаты.

В то же время методика SETIN достаточно универсальна, не привязана к конкретным видам языков программирования, учитывает современные требования по этапам разработки и способам программирования (объектно-ориентированное программирование), применима на ранних этапах разработки, при различных моделях жизненного цикла разработки, не требует привлечения экспертов из числа программистов, однако необходимо наличие качественно выполненной спецификации требований к ПО.

## ЛИТЕРАТУРА

1. Обзор метода функциональных точек [Электронный ресурс]. – Режим доступа: [http://citforum.ru/SE/project/arkhipenkov\\_lectures/12.shtml#6.4](http://citforum.ru/SE/project/arkhipenkov_lectures/12.shtml#6.4). – Дата доступа: 27.04.2020.
2. Шанченко, Н.И. Оценка трудоемкости разработки программного продукта: методические указания / Н. И. Шанченко. – Ульяновск: УлГТУ, 2015. – 40 с.
3. Основы методики СОСОМО II. [Электронный ресурс]. – Режим доступа: [http://citforum.ru/SE/project/arkhipenkov\\_lectures/13.shtml#6.5](http://citforum.ru/SE/project/arkhipenkov_lectures/13.shtml#6.5). – Дата доступа: 07.05.2020.
4. Укрупненные нормы затрат труда на разработку программного обеспечения. [Электронный ресурс]: постановление Министерства труда и социальной защиты Республики Беларусь 27.06.2007 № 91. – Режим доступа: <http://pravo.levonevsky.org/bazaby11/republic19/text861.htm>. – Дата доступа: 17.05.2020.
5. Методика расчета затрат на создание, развитие и сопровождение объектов информатизации государственных органов [Электронный ресурс]: приказ министра цифрового развития, инноваций и аэрокосмической промышленности Республики Казахстан от 27 июня 2019 года № 140/НК. – Режим доступа: <https://egov.kz/cms/ru/law/list/V1900018927>. – Дата доступа: 27.05.2020.

*The article presents the results of a scientific study where authors analyzed methods for assessing the complexity of software development, identifying their advantages and disadvantages, identifying the most suitable for use, taking into account adaptation for the Belarusian practice of forming a state order to carry out developments in the field of information technology.*

**Keywords:** labor input, software, metrics, methods, labor cost estimation.

Получено: 12.06.2020.